# Two-stage Framework for a Topology-Based Projection and Visualization of Classified Document Collections

Patrick Oesterling   Gerik Scheuermann*   Sven Teresniak   Gerhard Heyer†
University of Leipzig

Steffen Koch   Thomas Ertl‡   Gunther H. Weber§
University of Stuttgart   Lawrence Berkeley National Laboratory

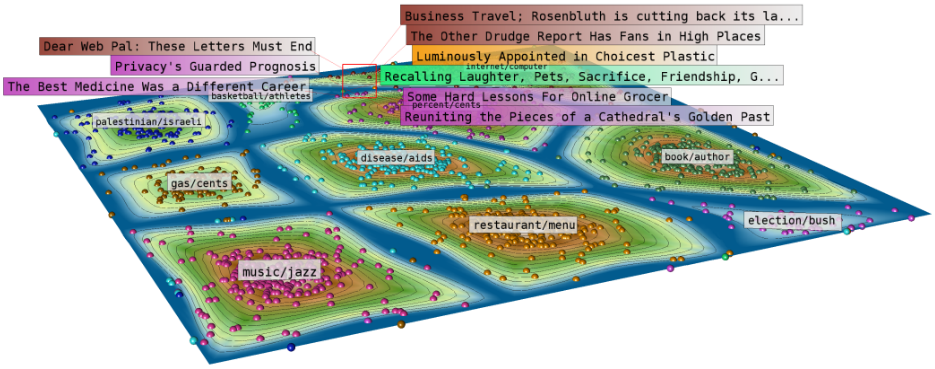

Business Travel; Rosenbluth is cutting back its la...
The Other Drudge Report Has Fans in High Places
Dear Web Pal: These Letters Must End
Luminously Appointed in Choicest Plastic
Privacy's Guarded Prognosis
internet/computer
Recalling Laughter, Pets, Sacrifice, Friendship, G...
The Best Medicine Was a Different Career
basketball/athletes
Some Hard Lessons For Online Grocer
percent/cents
Reuniting the Pieces of a Cathedral's Golden Past
palestinian/israeli
disease/aids
book/author
gas/cents
restaurant/menu
election/bush
music/jazz

Figure 1: Island-like visualization of a document point cloud's topological structure. By sharing similar dimensions, documents accumulate in subspaces of the high dimensional *information space.* Considering dimensions as words, clusters are assumed to describe topics, i.e., islands, in the final visualization.

## Abstract

During the last decades, electronic textual information has become the world's largest and most important information source available. People have added a variety of daily newspapers, books, scientific and governmental publications, blogs and private messages to this wellspring of endless information and knowledge. Since neither the existing nor the new information can be read in its entirety, computers are used to extract and visualize meaningful or interesting topics and documents from this huge information clutter.

In this paper, we extend, improve and combine existing individual approaches into an overall framework that supports topological analysis of high dimensional document point clouds given by the well-known *tf-idf* document-term weighting method. We show that traditional distance-based approaches fail in very high dimensional spaces, and we describe an improved two-stage method for topology-based projections from the original high dimensional *information space* to both two dimensional (2-D) and three dimensional (3-D) visualizations. To show the accuracy and usability of this framework, we compare it to methods introduced recently and apply it to complex document and patent collections.

**Index Terms:** H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Theory and methods; I.5.3 [Pattern Recognition]: Clustering—Algorithms;

## 1 Introduction

Due to the exponential increase of electronic textual data, computers are required to reasonably pre-process, classify and visualize coherent parts of massive data to prevent the user from aimlessly wandering through information clutter. For this purpose, researchers have been constantly searching for optimal models to accurately represent complex linguistic relationships. One of these models is the *vector space model* that represents documents as high dimensional vectors. In this paper, we propose a framework that makes it possible to investigate and visualize such a document point cloud using a topological approach. We focus on classified documents, instead of finding the classification itself, because classification is a discipline in its own right with many different approaches. Furthermore, most document collections, such as newspapers or patent collections, have their own classification schemes which are manually assigned. However, the classification itself does not support a meaningful data layout in lower dimensions. We, therefore, just use classification information to back up a topology-based projection from very high dimensions into three dimensions in order to create a visualization which describes the nesting structure and neighborhood relationship between the parts of the input point cloud. In order to avoid confusion regarding topological issues, we point out that we do not try to identify exact topological features, topological classes or homeomorphic spaces. Those things are considered in the field of algebraic or point-set topology whose algorithms and assumptions are way to complex to be applicable in very high dimensions. A good survey through this field is given in [2]. Instead of directly approximating a point set's or manifold's topology, we indirectly construct and study a scalar function's topology by means of the join-tree, which is rather limited to describe topological complex features. We, therefore, do not expect the user to be a topologist and understand ourselves as clustering competitors, since, as

*e-mail: {oesterling, scheuermann}@informatik.uni-leipzig.de
†e-mail: {teresniak, heyer}@informatik.uni-leipzig.de
‡e-mail: {Steffen.Koch, Thomas.Ertl}@vis.uni-stuttgart.de
§e-mail: ghweber@lbl.gov

we will see later on, neither the join-tree nor the 3-D landscape visualization are capable of visualizing various and complex high dimensional features. Regarding visual analytics aspects, we present the user, e.g. a journalist, a framework to obtain a 2-D/ 3-D layout of a set of documents. To achieve this, we use a topology-based projection which critically depends on a single parameter which has to be found by interacting with the visualization. This interactive analysis [17] supports the user to iteratively find the desired parameter, and therefore the desired information. In the end, since clustering information leads to only coarse insights, we deem our data layout as an initial point for further exploration.

## 2 Related Work

Text classification, as a mixture of information retrieval, machine learning, and (statistical) language processing, is concerned with building systems that partition an unstructured collection of documents into meaningful groups [20]. The two main variants are *clustering*, i.e., finding a latent structure of a previously determined number of groups, and *categorization*, i.e., structuring the data according to a group structure known in advance. For the latter *supervised* approach, different types of learners have been used, including probabilistic 'naive Bayesian' methods, decision trees, neural networks and example-based methods. However, in the context of computational learning theory and comparative experiments, support vector machines [12] and boosting [19] have been the two dominant learning methods in text classification in the recent years.

For structural insights of reasonably high dimensional data, one can directly visualize the point cloud using axis-based methods like, e.g., scatter plot matrices [1] or parallel coordinates [10]. Because these techniques depend on the data's dimensionality, it is often beneficial to previously project the data to lower dimensional spaces. By defining *meaningful* criteria, like, e.g., maximal variance or maximal distance between cluster centroids, projections try to preserve the structure in the projected dimension. Some of them are either supervised, such as LDA [6] or OCM [11], or unsupervised, such as PCA [13] or MDS [16]. Choo et al. [4] combined the advantages of several projections to minimize the loss of information during the transformation. Besides these linear projections, also non-linear embeddings exist which use additional structural information when determining a layout of the data in lower dimensions: Takahashi et al. [22] proposed a manifold learning approach to obtain a layout in 3-D which reflects the topology of a high dimensional input scalar function. They use the $k$-nearest-neighborhood graph to seek the manifold proximity and they define an own scalar-based distance measure to determine the closeness of points. In order to use this method for point cloud analysis, one has to ensure the definition of a suitable scalar function. For, e.g., clustering purposes, a meaningful scalar function should also be defined in the void part of a data set (where there are no vertices) in order to appropriately separate dense regions from regions of low density. Therefore, Oesterling et al. [17] focus on the appropriate construction of a point set's scalar function, supported by a neighborhood description by means of the Gabriel graph [7], instead of deriving a manifold from the function. In the end, their 3-D data layout reflects the topology of the data's approximated density function, realized by the *topological landscape* metaphor [24], a 3-D terrain which has the same topology as the input data set. Specifically for document collections, ThemeScape [25] also uses a terrain metaphor to visualize the data, however, utilizing different underlying models.

## 3 Background

The original contribution of this work lies in extending, improving and combining several individual approaches into an overall framework for analyzing document collections. To make it easier to the reader to follow the details in the later part of this paper, we introduce the most important concepts in this section.

### 3.1 Linear Discriminant Analysis (LDA)

For classified, high dimensional data, Choo et al. [4] described the dimension reduction as a trace optimization problem. Following their nomenclature, the clustered $m$-dimensional data points $a_i \in \mathbb{R}^m$ are given as a data matrix

$$A = [A_1 \;\; A_2 \;\; \cdots \;\; A_k], \text{ where } A_i \in \mathbb{R}^{m \times n_i} \text{ and } \sum_{i=1}^{k} n_i = n.$$

The groups of column vectors in $A$ correspond to the $k$ groups of $n$ clustered $m$-dimensional input vectors. For these groups $N_i$ of column indices of vectors belonging to cluster $i$, the cluster centroids $c^{(i)}$ and the global centroid $c$ are given by

$$c^{(i)} = \frac{1}{n_i} \sum_{j \in N_i} a_j \text{ and } c = \frac{1}{n} \sum_{j=1}^{n} a_j$$

Using the clusters' vectors together with their centroids and the global centroid, the within-cluster scatter matrix $S_w$ and the between-cluster scatter matrix $S_b$ are defined as

$$\begin{aligned} S_w &= \sum_{i=1}^{k} \sum_{j \in N_i} (a_j - c^{(i)})(a_j - c^{(i)})^T \\ S_b &= \sum_{i=1}^{k} n_i (c^{(i)} - c)(c^{(i)} - c)^T \end{aligned}$$

By calculating the trace of these scatter matrices as

$$\text{trace}(S_w) = \sum_{i=1}^{k} \sum_{j \in N_i} \|a_j - c^{(i)}\|_2^2 \tag{1}$$

$$\text{trace}(S_b) = \sum_{i=1}^{k} n_i \|c^{(i)} - c\|_2^2 \tag{2}$$

Choo et al. [4] specify cluster quality measures by considering the distances between the $k$ cluster centroids and the variance within each cluster, respectively. That is, well-separated clusterings usually will have a large trace$(S_b)$ and a small trace$(S_w)$. Eq.(1) describes trace$(S_w)$ as the squared sum of pairwise distances between a cluster's points and its centroid. Likewise, Eq.(2) describes the pairwise distances between the cluster centroids and the global centroid.

The fundamental idea of their approach is to consider dimension reduction as a trace optimization that maximizes trace$(G^T S_b G)$ and minimizes trace$(G^T S_w G)$ in the reduced dimensional space, using a dimension reducing linear transformation

$$G^T \in \mathbb{R}^{l \times m} : x \in \mathbb{R}^{m \times 1} \to z = G^T x \in \mathbb{R}^{l x 1}$$

projecting an $m$-dimensional input vector to an $l$-dimensional space $(m > l)$. It turns out that the solution, $G_{LDA}$, of the LDA criterion as

$$J_{b/w} = \text{maxtrace}((G^T S_w G)^{-1} (G^T S_b G))$$

consists of the column vectors which are the leading generalized eigenvectors $u$ of the generalized eigenvalue problem

$$S_b u = \lambda S_w u \tag{3}$$

and that LDA preserves the original cluster structure after projecting the $m$-dimensional input vectors into the $l$-dimensional space, such that $l = k - 1$. We refer the interested reader to reference [4] for further information, as it explains all the relationships clearly and in much more detail. In summary, LDA uses the additional clustering information of the input data to do a supervised projection from the original high dimensional space into an optimal lower dimensional space, i.e., $(k-1)$-dimensional, maximizing the inter-cluster distances and minimizing the intra-cluster distances in the reduced dimensional space.

### 3.2 Approximating a Point Cloud's Topology

Although the LDA preserves the clustering structure in the intermediate space in terms of its criteria, the target dimensionality might still be significantly larger than two or three. As a consequence, a subsequent projection will cause a loss of information due to projective overplotting in conventional visualizations. In order to avoid this second projection error, the intermediate space could be analyzed directly, instead of considering the point cloud in either the original $m$-dimensional space or in the 2-D/ 3-D space. Oesterling et al. [17] describe a method to analyze a point cloud's structure in higher dimensions followed by a topology-based projection to obtain a 3-D layout of the data which describes the structure in the original space. The basic idea is to describe the point cloud's structure indirectly by constructing a scalar function which appropriately reflects the data distribution in terms of density. In the context of density-based clustering, they have to evaluate the neighborhood of the given points in order to distinguish regions of both high and low density. Afterwards, they perform a topological analysis on this scalar field, utilizing the join-tree [3] which encodes the (joining-) evolution of contours, i.e., regions of equal density throughout the scalar function. The final visualization, in the end, reflects that join tree's hierarchical structure, which, in turn, reflects the structure of the point cloud's density distribution. In particular, they perform the following steps:

1) to facilitate the investigation of a point's neighborhood, the input point cloud is connected by the Gabriel graph [7] which is a special neighborhood graph

2) utilize the neighborhood graph to perform kernel density estimation at *meaningful* positions in space. In this case, at the graph vertices, i.e., at the data points (where it is dense), and on the mid-points of the graph's edges, i.e., in the void between two neighbored points (where it is likely not dense)

3) the join-tree computation is performed on this approximated density distribution to analyze the amount, size and (joining-) behavior of the contours. Because the contours describe the dense regions, this equals the determination of number, size and hierarchy of clusters

4) make use of the *topological landscapes* metaphor, proposed by Weber et al. in [24], to create a 3-D terrain which has the same topology as the join-tree. In this landscape, the structure of hills corresponds to that of the clusters in the data set

We show topological landscapes and variations of this metaphor throughout the next sections. Due space limitations, we refer to [17] and [24] for further information, since they introduce and explain a number of other concepts unrelated to the contributions of this work.

### 3.3 Volatility

In the *vector space model (vsm)* [18], text documents are represented as vectors. Each dimension corresponds to a separate term[1] and denotes the term's relevance in this particular document. Many different criteria have been proposed to extract only *meaningful* terms together with their individual significances. A common approach is the *if-idf* document-term weighting: for each single document, each term's frequency is weighted relatively to the number of other documents containing this term. Instead of only considering term frequencies in a single document or the whole corpus, additional semantic analysis can contribute to a vector's final expressiveness. Teresniak et al. [23, 9] proposed an approach to define a term's meaningfulness or topical relevance by looking at the temporal fluctuation of a term's global context (i.e., how neighboring terms change over time). Utilizing stock-market nomenclature, the authors call this fluctuation of context the term's *volatility*. Thus, analyzing the variation of a term's context for different time slices can be utilized to detect highly discussed topics and their importance over time. To provide rough outline of the method, the major calculation steps are:

1) Compute the significant co-occurrences $C(t)$ for each term $t$ in the whole corpus

2) Compute the significant co-occurrences $C_{T_i}(t)$ for each term $t$ in every time slice $T_i$

3) For every co-occurrence term $c_{t,j} \in C(t)$ compute $rank_{c_{t,j}}(i)$, the series of ranks of $c_{t,j}$ in the context of term $t$ in every time slice $T_i$

4) Compute the coefficient of variation (i.e., the ratio of the standard deviation to the mean) $CV(rank_{c_{t,j}}(i))$ for every co-occurrence term $c_{t,j} \in C(t)$

5) Compute the term's volatility as the average of these variances:
$$Vol(t) = \frac{1}{|C(t)|} \sum_j CV(rank_{c_{t,j}}(i))$$

When plotting a term's frequency and volatility over time, both quantities do not necessarily correlate, as the basic idea of volatility is to detect a topic (as a change of contexts) and not detecting just a heavy usage of high-frequent terms describing it. Although other methods may be used to increase the expressiveness of terms, we chose this model to additionally support the tf-idf measure and omit terms which are not volatile enough. For more details and examples comparing frequency and volatility, we refer to [23, 9].

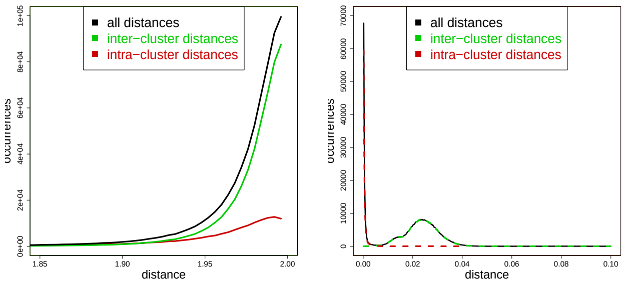
## 4 Problems in Very High Dimensions

It was Richard Bellman who first stated almost fifty years ago that *"a malediction has plagued the scientist from the earliest days"*. While this malediction, basically, concerns the problems caused by increasing the number of independent variables in different fields of application, especially for (metric) spaces, this means an exponential increase of volume with each additional dimension. As a consequence, particularly for distance-based approaches, it has be shown [21, 14, 8] that depending on the chosen metric, the distances between points either depend on the dimensionality ($L_1$ norm) or approach a constant ($L_2$ norm) or zero ($L_{d \geq 3}$ norm). That is, the relative difference between the distances to a point's farthest and closest point approaches zero. As a consequence, distances become relatively uniform in higher dimensions and some distance-based relationships such as *nearest neighbors* become meaningless in those spaces. Of course, if distances become uniform, every distance-based approach is affected by this phenomenon. To illustrate this problem for clustering algorithms, we consider the MEDLINE[2] data set. This data set consists of $1,250$ vectors in a $22,095$-dimensional space, divided into five equally sized clusters. The black graph in Figure 2(a) shows the number of individual distances between any two points. It is clearly visible[3] that the maximum of all distances lies around $2.0$ and $98,79\%$ of the distances are greater than $1.85$. The key issue is that both the inter-cluster distances (green) and the intra-cluster distances (red), which are obtained by considering the given clustering information, show the same behavior. If a data set contains several clusters, however,

[1] We take a term to mean the inflected type of a word, whereas a word is assumed to mean an equivalence class of inflected forms of a base form

[2] We use the sparse matrices kindly provided in [4]

[3] all diagrams can be arbitrarily magnified in the electronic version of this paper



(a) (b)

Figure 2: Partitioning of the MEDLINE distances between any (black) two points into inter-cluster (green) and intra-cluster (red) distances, done (a) in the original dimensionality and (b) after applying LDA. Now, the red and the green peak are separated.

this graph typically shows two peaks: one for the intra-cluster distances, and a peak representing the average distance between points [21]. Consequently, because only one peak is present, any purely distance-based approach will have issues with finding the underlying clustering of this data set.

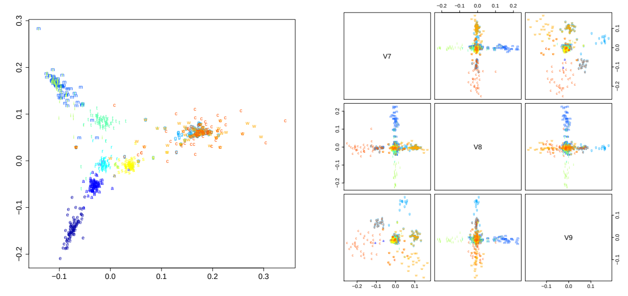
### 4.1 Concerning High Dimensional Documents

In addition to geometric issues, other semantic problems contribute to the measured distance between documents. If documents are represented by vectors, *meaningful* words usually serve as the vectors' dimensions. Although both a word and its significance heavily depend on the chosen algorithm, there will always be some words in the vector which disturb the basic assumption that (dis)similarity between documents is reflected by the distance between their corresponding vectors. Such words, like common and frequent words, can make two documents from different topic areas appear more similar than actually desired (or needed). The situation becomes worse if the distance between two documents is dominated by the contribution of non-discriminating words. It is possible that two document vectors about the same topic consist of equal numbers of discriminating and common words. If the common words are even distinct, they can cause the vectors to unintentionally disperse, thus negatively affecting the clustering.

In summary, when we consider a document collection as a point cloud, we are faced with two main problems. First, the construction of a point cloud where distances between points reflect (dis)similarities between documents. Second, due to curse of dimensionality, we are most likely not able to distinguish between similarity and dissimilarity, because most of the inter- and intra-cluster distances are uniform. To alleviate the first problem, language processing is necessary to avoid choosing meaningless words which force documents to unintentionally approach or disperse. For the second problem either a supervised (distance-independent in our case) clustering algorithm or a supervised projection to a lower dimensional space, being less afflicted by curse of dimensionality, is needed.

## 5 Two-stage Projection

Our two-stage approach is related to that proposed in [4]. Choo et al. describe several two-stage combinations of supervised LDA or OCM and unsupervised PCA to project the original input point cloud into an intermediate space, followed by a second projection down to 2-D. The supervised first stage projects the point cloud preserving its cluster structure and the goal of the second projection is to minimize information loss due to the dimension reduction to 2-D. One of the main contributions of our work is to improve the output of this two-stage approach by substituting the second stage a with direct topological analysis of the intermediate space. To motivate this, we first consider the two-stage LDA-LDA projection.



(a) (b)

Figure 3: (a) Rank-2 LDA of the REUTERS data set. Some clusters are still mixed in the projection (b) part of the 9-D scatterplot matrix, showing the intrinsic separation of the overplotted clusters.

### 5.1 Examining the Rank-2 LDA projection

This projection proposed by Choo et al. [4] consists of two subsequent LDA projections. The first from the original $m$-dimensional space into an intermediate $(k-1)$-dimensional space. The second LDA projects to $(k-1)$-dimensional data down to the 2-D space. As accurately explained in [4], the final projection matrix $V$,

$$V^T \in \mathbb{R}^{2\times m} \ : \ x \in \mathbb{R}^{m\times 1} \rightarrow z = V^T x \in \mathbb{R}^{2\times 1} = [u_1 \ \ u_2]$$

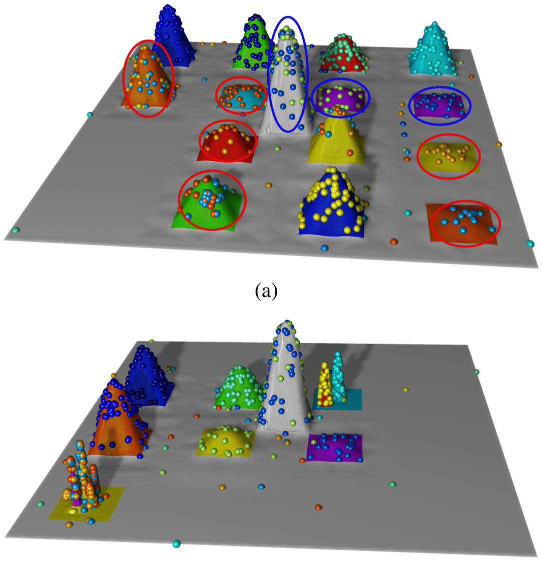
composing the two single projections, consists of the leading generalized eigenvectors of Eq. (3). Since the Rank-2 LDA and LDA+PCA are claimed to produce the best discriminating and almost identical results, we consider the Rank-2 LDA output, using the REUTERS[2] data set. This document collection consists of 800 vectors in a $11,941$-dimensional space, assigned equally to the following $k = 10$ clusters (the letters are used in the diagrams):

> earn ('e'), acquisitions ('a'), money-fx ('m'), grain ('g'), crude ('r'), trade ('t'), interest ('i'), ship ('s'), wheat ('w'), and corn ('c')

The result of the Rank-2 LDA is shown in Figure 3(a). As can be seen[3], the clustering is preserved well by the Rank-2 LDA. However, the clusters on the right-hand side and in the top left-hand corner of the scatter plot are overplotted. The pivotal question is why. Overplotting could either be due to data relationships, i.e., clusters are indeed mixed in the original space, or due to overplotting in the second stage. Since LDA preserves the cluster relationship in the $l$-dimensional space, we can analyze this 9-dimensional intermediate space by looking at a scatter plot matrix. Figure 3(b) shows us that the second assumption is true. Examining the point cloud not only from the first two principal components, but considering the $7^{th} - 9^{th}$ dimensions in the scatter plot matrix, it can be seen that both overplotted clusters consist of actually separated clusters in the intermediate vector space. This is not completely surprising, as the second-stage dimension reduction only uses two axes to discriminate the classes which contribute most to the second stage criteria. But nevertheless, due to the lack of any information about the intermediate space the user will most likely tend to (mistakenly) think of mixed clusters.

### 5.2 Substituting the Second Stage

To eliminate the drawback of overplotting clusters, we substitute the projection in the second stage with a topology-based projection from the $l$-dimensional intermediate space to a 3-D topological landscape (which can be easily reduced to a 2-D visualization


(a)

(b)

Figure 4: (a) topological landscape of the REUTERS data set in the intermediate 9-dimensional space. The topological analysis reveals the separated clusters which were overplotted by the Rank-2 LDA (b) without rebalancing the branch decomposition, the spacial relationship between density attractors can be read off the landscape.

as described in the next section). Considering the LDA-projected $(l = k-1)$-dimensional point cloud, at first the Gabriel graph is calculated to obtain neighborhood information, and a Gaussian-like filter kernel is applied on the graph's vertices and edges. Having this approximated density distribution, we determine its contour tree, or more precisely the *join-tree* as its representative. Subsequently, this tree serves as the input for the mapping process to achieve the final topological landscape visualization.

To justify this substitution of the second stage, we revisit the REUTERS example from above. Although the documents belonging to (c)orn, (w)heat and (g)rain are semantically related, the scatterplot matrix shows us that the corresponding clusters are still separated. These separated point accumulations lead to several density maxima inside the density distribution, resulting in several hills in the topological landscape. Figure 4(a) shows the topological landscape of the REUTERS data set. As can be seen, we are now able to distinguish dense parts of the point cloud which are separated by a region with low density. The hills of the landscape describe the topology of the point cloud's density distribution, i.e., hills correspond to contours evolving between their appearing at a density maximum and their merging at a saddle point. The colors of the hills have no special meaning and are chosen randomly. The small spheres correspond to the actual data points and are placed on the hills that correspond to their clusters. The colors of the spheres reflect their class association, thus corresponding to the coloring in Figure 3. As described in [17], the visual analysis process is performed by finding an appropriate filter radius. For this purpose, the user *examines* the landscape in each iteration and determines the filter radius for the next iterative step. The analysis process is finished when the user has extracted the desired clustering information or when the landscape denotes that the filter radius is getting too small. In the latter case, noise starts to produce density attractors or several extrema are found inside one cluster. Concerning the rebalancing, we point out that although this step was originally proposed in [24] to improve space utilization, for clusterings, the non-rebalanced landscape accurately reflects the spatial relationship between groups of points. As demonstrated in Figure 4(b), the hills (of each hierarchy level) are positioned in a spiral layout around the center hill, which corresponds to the global maximum. That is, the global maximum lies inside the 'm'/'i'-cluster, as this is the densest cluster (i.e., points per area). In the neighborhood to this density maximum there are two other attractors, corresponding to (local) accumulations of 'i'-class and 'm'-class points. Next to these clusters, the yellow 's'-cluster and the cyan 'r'-cluster are closer to each other than to the yet found 'i'/'m'-clusters. The same holds for the 'a', 'e' and 't' clusters, which all have their own density attractor, but have their saddle positions in the neighborhood of the clusters that have been found yet. The hilly nesting structure of the 'g'/'w'/'c' accumulations corresponds to a locally neighbored situation of these points, being separated from the remaining density maxima (clusters) by low density regions. A comparison of this landscape with Figure 3(a) leads to roughly the same neighborhood description, except for all the information that was lost by the second stage projection in Figure 3(a).

## 6 Extending the Visualization

Although it is capable of visualizing arbitrary, high dimensional data, the original topological landscape metaphor was only applied to visualize the topology of 3-D scientific scalar fields, given on regularly sampled grids. Using this metaphor to visualize data on a completely unstructured grid, sampled mostly inside the clusters and hardly in between, leads to some perceptual problems:

### 6.1 3-D to 2-D

First of all, as the landscape is still three dimensional, it suffers from overlapping of the hills and therefore the benefit of the visualization is view-dependent (especially when data points are positioned at the back of a hill). To alleviate the overlapping of hills and data points, we propose a flattening of the original topological landscape. Using the same construction scheme like in 3-D, we create a flat 2-D landscape by using the join tree's (interpolated) isovalues as additional vertex information instead of considering them as 3-D height values. On this 2-D scalar field, which has the same topology like the input join tree, we apply normal color mapping and isoline extraction to encode the absolute densities. In order to support the advantage of metaphors, we relate this visualization to an atoll by applying a color coding from blue (water) to yellow (beach), then fading from green (grass) into brown (mountains) and finally to white (snowy mountain top). The isolines, which correspond to the original height values, allow for an easier density correlation between the data points. Altogether, this visualization supports the same topological insights, but with far less overlapping in 2-D.

### 6.2 Improved Volume distribution

The second problem concerns the representation of approximated contour volumes, i.e., the size of clusters in our case. As described in [24], a metric-based distortion can be applied to the landscape in order to better reflect the real size of hills (contours), which otherwise would only depend on the landscape's construction scheme itself. Therefore, the triangles of a hill are resized according to the hill's corresponding cluster volume. Because this volume is distributed equally to all the triangles of a hill, the centered hill of nested hierarchies gets heavily distorted. This primarily destroys the visual expressiveness of the hill's corresponding cluster in the landscape. To solve this, we change the triangles' volume assignment. Instead of dividing a branch's volume by the number of the corresponding triangles, we assign the volume above the first saddle to the eight triangles of the centered hill and the volume beneath the first saddle is assigned equally to all the remaining triangles. Although this could be done more accurately, by considering each volume between each pair of saddles, we believe that this distribution

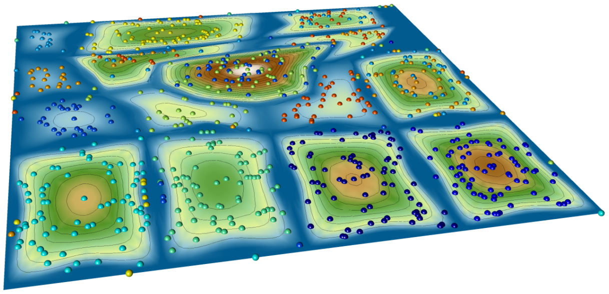


Figure 5: The flattened and volumetric distorted topological landscape of the REUTERS data set. The height lines and the coloring reflect the original height values (i.e., the absolute density values) from low (blue) to high (white). Furthermore, the distorted islands better reflect the actual cluster volumes.

sufficiently points out the volume of the corresponding cluster. Figure 5 shows the flattened REUTERS landscape with the size of the hills (islands) corresponding to the clusters' sizes (approximated by the number of points).

### 6.3 Labeling

We additionally enhance this visualization with a labeling of both the hills and the small data spheres. Since we are dealing with documents, we most likely want to show their titles at first. But showing the titles of all documents would result in a huge overplotting of the labels. Therefore, we have implemented the basic 'excentric labeling' approach, proposed by Fekete et al. [5]. In this approach the user slides a small focus area over the data set, labeling only those data points which lie inside the small focus window. The labels are positioned around the focus area and they are connected with their corresponding data entities by a line. In the end, the labels are colored according to the class of the data points. If titles are too long, we cut them or use a given short version of the title.

Following the ideas of *tf-idf* and the *vector space model*, the clusters are thought to constitute topics. Document vectors belonging to the same topic most likely share similar words with similar significances. Therefore, these document vectors accumulate in the subspace spanned by the dimensions (words) they share. The goal is to provide hills (clusters) with labels corresponding to their topics. Instead of using the documents' class-association, we propose a quick semantic analysis. Considering the documents on a single hill, we search for the most frequent word(s) they share. Finally, we use the two most frequent words as a hill's topic.

## 7 Classification

Assuming an unclassified entity would match one of the given classes, it is possible to use our proposed two-stage process for classification purposes: The LDA, as a linear dimension reduction, projects similar high dimensional vectors into the same region in the lower dimensional space. Therefore, if we assume that we already have *learned* the LDA projection for a given class (based on the classified input data), the nature of LDA ensures that additional similar unclassified vectors are projected into the same lower dimensional target area. While this, on the one hand, is solely owed to the projection itself, we can, however, rely on the fact that similar vectors (classified or not) are comprised by the same contours of the density function. Therefore, although it is the first stage projection which ensures a clustering in the lower dimensional space, it is the second stage topological projection which eventually gives us a tool for detecting the clustering and to perform a classification, based on cluster association. From our algorithm's point of view, we assign unclassified data entities the class of their neighbored classified entities as follows: We determine the LDA projection matrix based on the classified data and use this matrix to project both the classified and unclassified data (resulting in accumulations in the lower dimensional space if the vectors are similar). We now have two choices: First, we combine both point clouds to serve as the input for the topological analysis, or we alternatively apply the topological analysis solely on the classified point cloud and approximate the topology of the combined point cloud as follows: We extend the height graph of the input point cloud by an edge between each unclassified data point and its nearest classified neighbor (this equals the finding of the nearest contour). Afterwards, we compute the densities at these edges' mid-points and at the unclassified points (in both cases, based on the classified entities). On this extended height graph, we continue as before. For both scenarios, we find similar classified and unclassified documents on the same branches in the branch decomposition, thus assigning an unclassified entity the label of the most frequent class on the entity's branch (i.e., island in the landscape).

## 8 Discussion and Examples

Analyzing the $l$-dimensional intermediate space is, of course, more expensive than analyzing a (lossy) 2D space. For detailed runtime complexities we refer to [4], [17] and [24] and discuss some common runtime issues instead: First, the prior LDA projection greatly accelerates the topological analysis of the point cloud's density distribution in the intermediate vector space. This acceleration is due to the fact that the reduced dimensionality leads to a less full Gabriel graph, resulting in less necessary density evaluations on the graph's edges. Since graph connectivity commonly increases exponentially with each additional dimension, this approach leads to significant difference for high dimensional data sets. Besides this, we can also use the classification ideas from the last section and treat a percentage of the input data as it was unclassified. By randomly using, e.g., only 50% of the input vectors to *learn* the LDA projection matrix, the remaining 50% of the vectors are thought to be projected correctly, due to their similarity to the training data. Of course, using only a part of the input data greatly accelerates the runtime of the LDA. We will demonstrate the quality of both classification and using only a subset of the data later in this section.

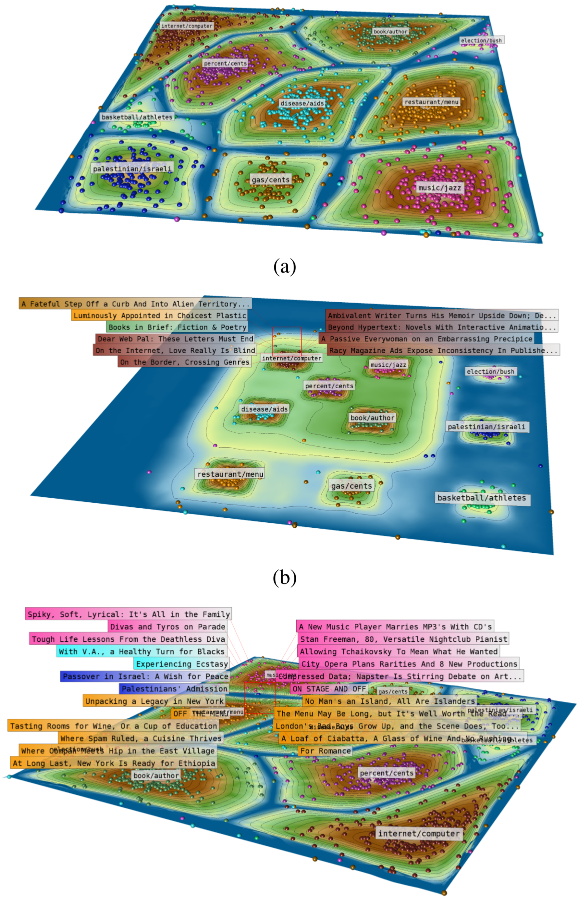
For demonstration purposes, we apply our method on a *New York Times* document collection and a patent collection. As mentioned, we refer to the literature for detailed information regarding the runtime of the individual steps. To provide a rough guideline, the topological analysis of the previous REUTERS example and the upcoming examples in this chapter took around four seconds each. Our machine has 8GB memory and we use two 2,6GHz-QuadCore processors to benefit from parallelism.

### 8.1 NYT - Document Collection

The *New York Times Annotated Corpus*[4] contains over 1.8 million articles written and published by the New York Times between January 1, 1987 and June 19, 2007 with article meta-data provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. As part of the New York Times' indexing procedures, most articles are manually summarized and tagged by a staff of library scientists.

For our testing purposes, we consider the year 2001 and extract 50 documents per day. As described in Section 3, we use the *tf-idf* document-term weighting and the terms' volatility to determine the document vectors. Therefore, considering a single document, all the stop words are pruned and the normal tf-idf weighting scheme is applied to the remaining words. Subsequently, a 30-day sliding window is used to determine a term's volatility for each of the 365 days. Then, we use the variance of each term's volatility series to achieve an ordered series of over-year importance of

[4] http://www.ldc.upenn.edu


internet/computer
book/author
election/bush
percent/cents
disease/aids
restaurant/menu
basketball/athletes
palestinian/israeli
gas/cents
music/jazz
A Fateful Step Off a Curb And Into Alien Territory...
Luminously Appointed in Choicest Plastic
Books in Brief: Fiction & Poetry
Dear Web Pal: These Letters Must End
On the Internet, Love Really Is Blind
On the Border, Crossing Genres
Ambivalent Writer Turns His Memoir Upside Down; De...
Beyond Hypertext: Novels With Interactive Animatio...
A Passive Everywoman on an Embarrassing Precipice
Racy Magazine Ads Expose Inconsistency In Publishe...
Spiky, Soft, Lyrical: It's All in the Family
Divas and Tyros on Parade
Tough Life Lessons From the Deathless Diva
With V.A., a Healthy Turn for Blacks
Experiencing Ecstasy
Passover in Israel: A Wish for Peace
Palestinians' Admission
Unpacking a Legacy in New York
OFF THE MENU
Tasting Rooms for Wine, Or a Cup of Education
Where Spam Ruled, a Cuisine Thrives
Where Oompah Meets Hip in the East Village
At Long Last, New York Is Ready for Ethiopia
A New Music Player Marries MP3's With CD's
Stan Freeman, 80, Versatile Nightclub Pianist
Allowing Tchaikovsky To Mean What He Wanted
City Opera Plans Rarities And 8 New Productions
Compressed Data; Napster Is Stirring Debate on Art...
ON STAGE AND OFF
No Man's an Island, All Are Islanders
The Menu May Be Long, but It's Well Worth the Read...
London's Bad Boys Grow Up, and the Scene Does, Too...
A Loaf of Ciabatta, A Glass of Wine And No Rushing...
For Romance

(a)

(b)

(c)

Figure 6: (a) 2-D visualization of the NYT data set. Islands correspond to topics and their sizes fit the clusters' number of documents. (b) The same landscape for a higher filter radius and without volumetric distortion. The small spheres between the islands are assumed to be outliers or noise (i.e., documents with too common vocabulary). (c) documents inside the focus area are labeled with their titles.

all volatile terms. Finally, we clip the tf-idf vectors by the words which are not assumed to be volatile enough (based on their position in the ordered list). As a classification, we randomly choose documents corresponding to 10 different tags, up to 250 per group. Altogether, this test case consists of $1,896$ documents (points), described by $46,393$ words (dimensions). Figure 6 shows some pictures of the final visualization. As can be seen, the LDA and the subsequent topology-based projection down to 2-D preserve the 10 topical clusters, successfully described by the vector space model. The proposed labeling of the hills appropriately reflects the underlying topics, suggested by the document titles and implied by the documents' content. Looking at the visualization, it is important to understand that closeness of islands does not imply that the corresponding topics are related in the original space. This information gets lost during the projection and, therefore, spatial relationships are only encoded in the hierarchy of hills. (i.e., only *sub-hills* express spatial closeness to their parent hill).

### 8.2 Patent Collection

Access to patent information is of importance for a variety of interest groups today. Besides many other properties, the majority of information describing the nature of a patent is still conveyed through its textual content, therefore making natural language processing (NLP) a mandatory part of solutions for patent analysis. The shear mass, complexity, high dimensionality, and heterogeneity of patent data, make scalable visual analytics approaches for patent analysis [15] a hard task. One particularly relevant type of meta data that is available for patent applications is manually assigned classification information. This classification information organizes the vast numbers of patents into predefined classes representing certain technical or functional aspects. Several different schemes for patent classification, such as the International Patent Classification (IPC), Japanese F-terms, and the US classification, exist. Patent offices are interested in automatic classification of new patent applications according to the existing classification schemes.

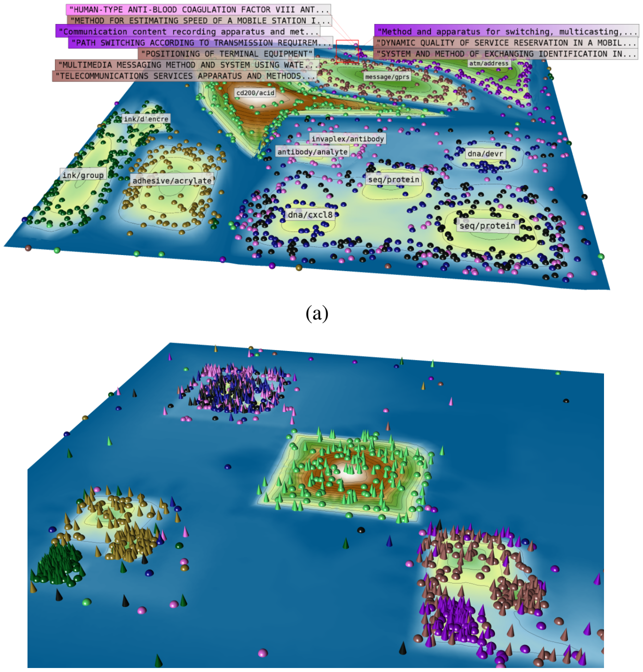
In order to evaluate our approach, we tested against the IPC comprising more than $70,000$ classes, hierarchically organized into sections, classes, subclasses, main groups, and sub groups. In the end, our test case consists of $1,552$ randomly selected patents from different IPC hierarchies (up to 200 each):

'A61K..38/17', 'C12N...1/21', 'H04Q...7/22', 'B41C...1/10',
'C09D..11/00', 'C09J...7/02', 'G01N..33/53', 'H04Q..11/04'

We used patent data[5] from the European Patent Office (EPO). As a preprocessing step, the data has been analyzed and the text content was stored in vectorized form within a search index. From this index the tf-idf values for all dimensions of the term vectors have been computed. First, we examine whether our landscape reflects the nesting structure of the chosen patent hierarchy. Figure 7(a) shows the visualization for our test case. Mainly four groups can be identified: The purple and brown points in the upper-right corner, belonging to H04Q patents, clearly address networking, as the labels (and the document titles) relate to *atm, address, message, ip and cell*. In fact, the H04Q IPC-hierarchy categorizes patents belonging to "electricity" (H), "electric communication technique"(H04) and "selecting (switches, relays, etc)" (H04Q). Although the group of pink, blue and black points on the right-hand side belongs to completely different IPC-sections (A61K, C12N, G01N), the corresponding patents all concern medical issues in their major field: A - Human Necessities, C - Chemistry, G - Physics. Because they share the medical vocabulary, they still constitute one topic in the vector space model. Finally, the centered hill belongs to the B41C cluster and the green and golden points comprise a cluster related to applications of materials in chemistry and metallurgy (C09D, C09J).

We want to use this patent example to furthermore demonstrate the classification ideas from section 7. We, therefore, split the patents into 50% classified training data and 50% unclassified test data, which means that we henceforth ignore their class label. For illustration purposes, however, we remember the test data's class association for coloring in the landscape. After determining the training set's LDA projection matrix, we use it to project both patent sets and use their combination for our topological analysis. Figure 7(b) shows the (not volumetric distorted) landscape. As can be seen, the training data (represented by spheres) and the test data (represented by cones) belonging to particular classes (represented by the color) are all hosted on mainly their own islands. This nicely confirms, that due to their shared vocabulary, i.e., their shared dimensions, patents of a specific class are equally handled by the LDA, and their accumulation in the lower dimensional space allows us to topologically find the dense area as one combined cluster. While this allows for a faster LDA computation, the topological encoding by means of the join tree's branch decomposition also offers a way to provide the test data the (main) class of a branch's / hill's training data class. Regarding our example, the runtime of the LDA using all the data took 28*s*, whereas using only 50% took only 12*s*. Regarding the classification quality, we determined for each branch (island) how

[5] from 'Text of EP-A documents' and 'Text of EP-B documents'


"HUMAN-TYPE ANTI-BLOOD COAGULATION FACTOR VIII ANT...
"METHOD FOR ESTIMATING SPEED OF A MOBILE STATION I...
"Communication content recording apparatus and met...
"PATH SWITCHING ACCORDING TO TRANSMISSION REQUIREM...
"POSITIONING OF TERMINAL EQUIPMENT"
"MULTIMEDIA MESSAGING METHOD AND SYSTEM USING WATE...
"TELECOMMUNICATIONS SERVICES APPARATUS AND METHODS...
"Method and apparatus for switching, multicasting,...
"DYNAMIC QUALITY OF SERVICE RESERVATION IN A MOBIL...
"SYSTEM AND METHOD OF EXCHANGING IDENTIFICATION IN...
atm/address
message/gprs
cd200/acid
ink/d'encre
invaplex/antibody
antibody/analyte
dna/devr
ink/group
adhesive/acrylate
seq/protein
dna/cxcl8
seq/protein

(a)

(b)

Figure 7: (a) 2-D topological landscape of the patent data set. The nesting structure of the islands reflects the IPC hierarchy of the test data set. (b) The same landscape without volumetric distortion. Even by *learning* the LDA with only 50% of the input data, the remaining patents (the cones) are placed on their correct islands (clusters).

many of the branch's test data entities match the branch's training data class (using the test data's known class in this case). On average, 89.4% of the test data on a branch matches the class of the training data, or more precisely $\approx 76.7\%$ in the noisy region of the medicine archipelago, and $\approx 99.6\%$ on the remaining branches which correspond to clusters being better separated.

## 9 Conclusions and Future Work

In order to cluster document point clouds, we referred to the necessity of reflecting similarity by distance and we pointed to uniform distances, caused by the curse of dimensionality. We tried to alleviate the first problem by using a term's volatility, as we think that this results in more topical related terms. Concerning the dimensionality, we showed that a supervised approach is necessary in very high dimensional spaces. Therefore, we proposed a two-stage framework consisting of a supervised LDA projection down to $(k-1)$-D, followed by a direct topological analysis of this intermediate vector space. By doing so, we were able to improve comparable approaches which use lossy second stage projections. We also extended the visualization in [17] to facilitate a more precise and less overlapping analysis process in 2-D. For classification purposes, we showed how the LDA and the use of the branch decomposition can be used for automatic document classification based on an existent classification. Furthermore, the quality of the classification itself can be verified by examining the distribution of colored points in the landscape. If a single color occurs on several hills, the clustering (and therefore the classification) might be inappropriate. Since the presumed classification is some kind of a drawback compared to unsupervised approaches, our future work will concern the investigation and support of classification methods (maybe also topology-based). We also consider other data structures to identify more complex topological features.

## Acknowledgements

We thank Christian Heine for valuable remarks and inspiring discussions. We also thank the anonymous reviewers for their useful comments. This work was supported by a grant from the German Science Foundation (DFG), within the strategic research initiative on Scalable Visual Analysis (SPP 1335).

## References

[1] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[2] G. Carlsson. Topology and data. *Bulletin of the AMS*, 46, 2009.

[3] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 2003.

[4] J. Choo, S. Bohn, and H. Park. Two-stage Framework for Visualization of Clustered High Dimensional Data. In *IEEE VAST*, 2009.

[5] J.-D. Fekete and C. Plaisant. Excentric labeling: dynamic neighborhood labeling for data visualization. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, 1999.

[6] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[7] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[8] A. Hinneburg, C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? 2000.

[9] F. Holz and S. Teresniak. Towards automatic detection and tracking of topic change. In A. Gelbukh, editor, *Proc. CICLing 2010, Iaşi, LNCS 6008*. Springer LNCS, 2010. accepted for oral presentation.

[10] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 361–378, 1990.

[11] M. Jeon, H. Park, and J. B. Rosen. Dimensional reduction based on centroids and least squares for efficient processing of text data. In *Proceedings for the First SIAM Intl. Workshop on Text Mining*, 2001.

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, 1997.

[13] I. T. Jolliffe. *Principal component analysis*. Springer, Berlin, 1986.

[14] K. B. Jonathan, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *In Int. Conf. on Database Theory*, pages 217–235, 1999.

[15] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during patent search and analysis. In *Proc. IEEE Symp. Visual Analytics Science and Technology*, pages 203–210, 2009.

[16] J. B. Kruskal and M. Wish. *Multidimensional Scaling (Quantitative Applications in the Social Sciences)*. SAGE Publications, 1978.

[17] P. Oesterling, C. Heine, H. Jaenicke, and G. Scheuermann. Visual analysis of high dimensional point clouds using topological landscapes. In *IEEE Pacific Visualization 2010 Proceedings*, 2010.

[18] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.

[19] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning*, pages 135–168, 2000.

[20] F. Sebastiani and C. N. D. Ricerche. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.

[21] M. Steinbach, L. Ertöz, and V. Kumar. The challenges of clustering high-dimensional data. In *In New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recogn.*, 2003.

[22] S. Takahashi, I. Fujishiro, and M. Okada. Applying manifold learning to plotting approximate contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1185–1192, 2009.

[23] S. Teresniak, G. Heyer, G. Scheuermann, and F. Holz. Visualisierung von Bedeutungsverschiebungen in großen diachronen Dokumentkollektionen. *Datenbank-Spektrum*, 31:33–39, December 2009.

[24] G. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1416–1423, 2007.

[25] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. *Information Visualization, IEEE Symposium on*, 0:51, 1995.